UNIVERSIDAD DE COSTA RICA
SEDE DE OCCIDENTE
SECCION DE MATEMATICA

Viernes 23 de agosto
Programa
II Ciclo 2002

Programa del Curso
Ma–0560 Computación y métodos numéricos

0. Introducción:

   Es importante que el estudiante reconozca que método numérico usar en un determinado problema, que tenga una base matemática solida en cuanto a la obtención del método y que pueda programar los algoritmos.

   En este curso usamos MATHEMATICA como lenguaje de programación, la exposición magistral para obtener los diferentes métodos y el aporte de todos en el desarrollo práctico del curso.

1. Objetivos:

   1.1 Qué el estudiante conozca como obtener los principales métodos numéricos y sus aplicaciones.

   1.2 Qué el estudiante pueda leer y construir algoritmos para resolver problemas.

   1.3 Qué el estudiante pueda programar en MATHEMATICA.

2. Contenidos

   2.1 Preliminares Matemáticos: algunos teoremas, error, error absoluto, rapidez de convergencia, algoritmos estables e inestables.

   2.2 Solución de Ecuaciones: Algoritmo de bisección, punto fijo, Newton-Raphson. Análisi de error para métodos iterativos y técnicas de aceleración, ceros de polinomios y método de Müller.

   2.3 Interpolación y aproximación: polinomos de Taylor, interpolación y polinomio de Lagrange, Hermite, Spline cúbico.

   2.4 Integración numérica: extrapolación Richardson, integración numérica, integración numérica compuesta, integración Romberg, integrales múltiples.

3. Evaluación: un examen parcial (incluye las secciones [2.1], [2.2], [2.3]) con un valor de 30 %; participación en clase 10 %; trabajos, exposiciones y tareas programadas 60 %.

   Si nota es ≥ 7.0 aprueba el curso, si $6.0 \leq$ nota < 7.0 tiene derecho a ampliación y en otro caso pierde el curso.

4. Bibliografía

4.1 Burden y Faires. *Análisis numérico.* Grupo Editorial Iberoaméricana, 1985.

4.2 Krommer and Ueberhuber. *Computational Integration.* SIAM, 1998.

4.3 Zhongying Chen. *advances in computational mathematics.* Marcel Dekker, 1999.

4.4 Wolfram Stephen. *Mathematica.* Adddison–Wesley Publishing Co., 1991.

...Creo que es correcto afirmar que los criterios de selección del matemático, y también los de su éxito, son principalmente estéticos.
J. VON NEUMANN, 1947

Prof. Carlos Manuel Ulate Ramírez
dianau@racsa.co.cr

cmur.........

Algoritmo de bisección 2.1

Para encontrar una solución de $f(x) = 0$ dada la función continua f en el intervalo $[a, b]$ donde $f(a)$ y $f(b)$ tienen signos opuestos:

ENTRADA extremos a, b; tolerancia TOL; máximo número de iteraciones N_0.

SALIDA solución aproximada p o mensaje de fracaso.

Paso 1 Tomar $i = 1$.

Paso 2 Mientras que $i \leq N_0$ seguir Pasos 3-6.

Paso 3 Tomar $p = a + (b - a)/2$. (*Calcular p_i.*)

Paso 4 Si $f(p) = 0$ ó $(b - a)/2 < TOL$ entonces
SALIDA (p); (*Procedimiento completado satisfactoriamente*)
PARAR.

Paso 5 Tomar $i = i + 1$.

Paso 6 Si $f(a)f(p) > 0$ entonces tomar $a = p$ (*Calcular a_i, b_i*).
si no tomar $b = p$.

Paso 7 SALIDA ('El método fracasó después de N_0 iteraciones, N_0 = ', N_0);
(*Procedimiento completado sin éxito*).
PARAR.

Mencionaremos brevemente otros procedimientos de paro que pueden también aplicarse en el paso 4 del algoritmo 2.1, todos los cuales se aplican a cualquier técnica iterativa considerada en este capítulo. Seleccione una tolerancia $\varepsilon > 0$ y genere $p_1, \ldots, p_N$ hasta que una de las siguientes condiciones se satisfaga:

(2.1) $$|p_N - p_{N-1}| < \varepsilon,$$

(2.2) $$\frac{|p_N - p_{N-1}|}{|p_N|} < \varepsilon, \qquad p_N \neq 0, \text{ o}$$

(2.3) $$|f(p_N)| < \varepsilon.$$

Desafortunadamente, pueden surgir dificultades usando cualquiera de estos criterios de paro. Por ejemplo, existen sucesiones $\{p_n\}$ con la propiedad de que las diferencias $p_n - p_{n-1}$ convergen a cero mientras que la sucesión misma diverge. (Ver ejercicio 13). Es posible también que $f(p_n)$ esté cerca de cero mientras que p_n difiere significativamente de p. (Ver ejercicio 14). Sin conocimiento adicional acerca de f o p, la desigualdad (2.2) es el mejor criterio de paro que puede aplicarse porque verifica al error relativo.

Cuando usamos una computadora para generar las aproximaciones, es una buena idea añadir una condición que imponga un máximo al número de iteraciones

FIGURA 2.4

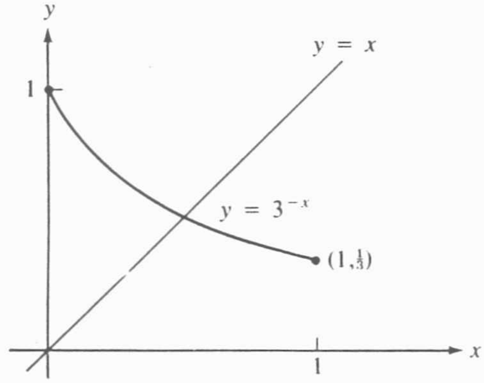
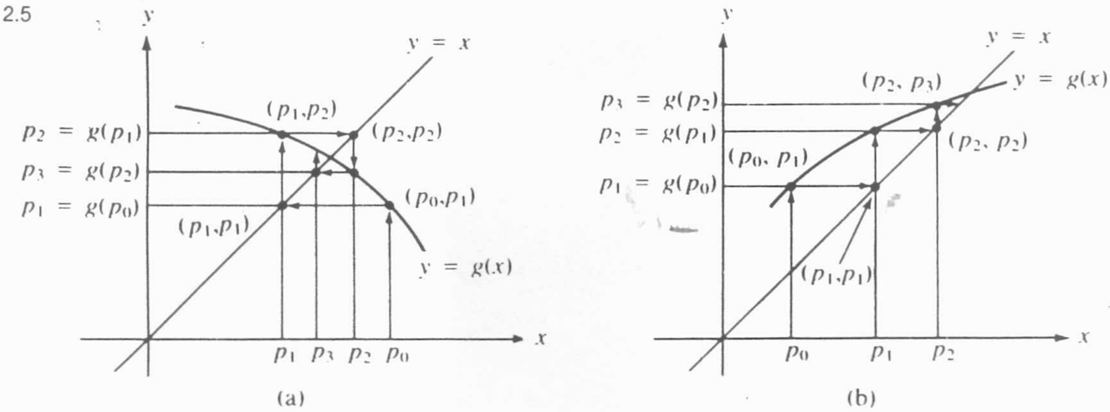


FIGURA 2.5



Algoritmo de punto fijo 2.2

Para encontrar una solución de $p = g(p)$ dada una aproximación inicial p_0:

ENTRADA aproximación inicial p_0; tolerancia TOL; número máximo de iteraciones N_0.

SALIDA solución aproximada p o mensaje de fracaso.

Paso 1 Tomar $i = 1$.

Paso 2 Mientras que $i \leq N_0$ seguir Pasos 3-6.

  Paso 3 Tomar $p = g(p_0)$. (*Calcular p_i*).

  Paso 4 Si $|p - p_0| < TOL$ entonces
  SALIDA (p); (*Procedimiento completado satisfactoriamente*).
  PARAR.

  Paso 5 Tomar $i = i + 1$.

  Paso 6 Tomar $p_0 = p$. (*Redefinir p_0*).

Paso 7 SALIDA ('El método fracasó después de N_0 iteraciones, N_0 =', N_0);
(*Procedimiento completado sin éxito*).
PARAR.